Universitat Oberta de Catalunya

uoc.edu

Servicio de firma de documentos almacenados en la nube

Gonzalo de Moya Sánchez
Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Sistemas de autenticación y autorización

Nombre Consultor/a: Juan Carlos Fernández Jara
Nombre Profesor/a responsable de la asignatura: Víctor García Font

28 de diciembre del 2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Servicio de firma de documentos almacenados en la nube*
Nombre del autor:	*Gonzalo de Moya Sánchez*
Nombre del consultor/a:	*Juan Carlos Fernández Jara*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	12/2020
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Sistemas de autenticación y autorización*
Idioma del trabajo:	*Castellano*
Palabras clave	*eIDAS, oAuth2, Seguridad*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Motivado por la nueva ley eIDAS que posibilita la firma electrónica cualificada de documentos en línea, se implementará un servicio de firma en la nube, con el que se podrán firmar documentos en línea, sin que haya ninguna diferencia a nivel legal con una firma física y que, además, no esté limitado por el soporte físico desde donde se esté haciendo la firma.

De forma más concreta, se habilitará la firma de PDFs almacenados en los proveedores de almacenamiento de documentos Dropbox y Google Drive, y se utilizará como proveedor de claves de firma a TrustedX.

De esta forma se explorarán las posibilidades y limitaciones que nos da la nueva regulación eIDAS.

Abstract (in English, 250 words or less):

Motivated by the new eIDAS regulation that enables the qualified electronic signature of documents online, it will be implement a signature service in the cloud, in which documents would be signed online, without any difference, at legal level, with a physical signature and it is not limited by the physical support from which the signature is being made.

More specifically, the PDFs stored in the document storage providers Dropbox and Google Drive, and TrustedX will be used as a provider of signing keys.

In this way, the possibilities and limitations of the eIDAS regulation will be explored.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En un mundo cada día más digital, resulta esencial poder autentificar la identidad del firmante de un documento digital de forma fácil, segura e inequívoca.

Los métodos usados en la actualidad tienen muchas deficiencias. Esto es comprensible por la importancia de las firmas digitales, y la dificultad de almacenar las claves de firma en un sitio seguro, ya que un robo de esta puede acarrear un grave problema legal.

El artículo 3.3 de la Ley 59/2003 define la firma electrónica cualificada como la firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma. Para que un dispositivo sea considerado como seguro, debe cumplir que las claves sean únicas y secretas, que la clave privada no se puede deducir de la pública y viceversa, que el firmante pueda proteger de forma fiable las claves, que no se altere el contenido del documento original y que el firmante pueda ver qué es lo que va a firmar [1].

Hasta hace unos pocos años, uno de los pocos dispositivos seguros reconocidos en España era el Documento Nacional de Identidad electrónico, DNIe por sus siglas. El DNIe acredita electrónicamente la identidad personal de su titular y permite la firma electrónica de documentos, esto es posible gracias a que incorpora un chip con los certificados necesarios para la firma digital. Otros muchos países también optaron por soluciones similares.

Este método de firma acarrea varios problemas, el principal es que consiste en una tarjeta física para la firma de un documento digital, lo que implica que haya un soporte físico para poder conectar el mundo físico con el digital, este soporte es el lector de tarjetas. El uso de este lector no es trivial ya que su uso conlleva la instalación de los *drivers* específicos en el equipo y la compatibilidad de estos *drivers* varía entre los distintos Sistemas Operativos.

Otro gran problema de este tipo de firma, que ha impactado la poca penetración de esta tecnología, es que, puesto que un ordenador personal no es un dispositivo seguro de creación de firmas, las firmas generadas son sólo firmas avanzadas según la definición de la ley.

Para tratar de solventar todos estos problemas, en el año 2014 se lanzó una nueva regulación en la Unión Europea llamada eIDAS, por sus siglas en inglés electronic IDentification, Authentication and trust Services, que puede dar solución a todos estos problemas. Esta nueva regulación permite que se disponga de la firma cualificada en distintos proveedores de servicios y se detallan los requerimientos de seguridad que los proveedores deben cumplir para obtener la certificación europea y figurar en la lista de confianza de proveedores de firmas electrónicas.

Con esta nueva normativa se abre todo un abanico de posibilidades y se actualiza a la nueva tendencia que está habiendo estos últimos años, donde todo está en la nube y no es necesario tener un dispositivo físico potente para tener todas las capacidades y prestaciones de este.

Un ejemplo de las nuevas oportunidades que ofrece esta nueva normativa la podemos encontrar en el anuncio de la próxima versión del DNI español, llamada 4.0, que permitirá llevar este documento identificativo directamente en el móvil [2].

Este Trabajo Final de Máster, TFM por sus siglas, está motivado por esta nueva ley y las nuevas oportunidades que ofrece. Durante este trabajo se explorarán las posibilidades y limitaciones que nos da la regulación eIDAS.

Se implementará un servicio de firma en la nube, con el que se podrán firmar documentos en línea, sin que haya ninguna diferencia, a nivel legal, con una firma física y que, además, no estemos limitados por el soporte físico desde donde se esté haciendo la firma.

Más concretamente, se habilitará la firma de PDFs almacenados en proveedores de almacenamiento de documentos como Dropbox y Google Drive, y se utilizará como proveedor de claves de firma a TrustedX.

1.2 Objetivos del Trabajo

El objetivo principal de este TFM es el estudio e implementación de una plataforma de firma digital de documentos PDF almacenados en servicios de alojamiento de archivos multiplataforma en la nube, tales como Google Drive o Dropbox, utilizando un proveedor de claves de firma certificado como es TrustedX. Los hitos para conseguir llegar a este objetivo final son:

- Estudio y comprensión de la nueva regulación eIDAS.

- Diseño e implementación de un sistema de identidades y autorización basado en OAuth2.

- Diseño e implementación de un servicio que integre las APIs de Google Drive y Dropbox, permitiendo el consumo y manipulación de documentos almacenados en estos servidores.

- Diseño e implementación de un servicio que integre al proveedor de claves de firma TrustedX de Entrust.

- Diseño e implementación de un servicio de firma de documentos, basándose en los servicios implementados en los puntos anteriores.

1.3 Enfoque y método seguido

El servicio que vamos a implementar, aunque está basado en muchos productos, será un desarrollo completamente nuevo que pretende demostrar las bondades de uso de un proveedor de firmas alojado en la nube frente a las alternativas utilizadas en la actualidad.

El proyecto tiene dos partes diferenciadas:

1. Parte teórica de investigación sobre las herramientas a utilizar, tecnología y regulación relacionada con el trabajo.
2. Parte práctica de implementación e integración de todos los componentes externos del sistema.

Para este trabajo usaremos una estrategia de mínimos, donde trataremos de conseguir el objetivo final del proyecto lo antes posible, y una vez ahí, se irá profundizando en cada una de sus partes, mejorándolo y haciéndolo más robusto y completo. De este modo se reduce la probabilidad de no conseguir el objetivo del proyecto, y también conseguimos unir la parte teórica y práctica del trabajo de forma gradual.

De este modo, se parte de un servicio de firma de documentos en local que utilice las firmas almacenadas en el proveedor de firmas TrustedX, con lo que se conseguiría el objetivo principal, y a partir de ahí, se irán integrando los demás componentes y objetivos secundarios (OAuth2, Dropbox) hasta llegar al producto final.

1.4 Planificación del Trabajo

Para la consecución del trabajo se seguirá una metodología *agile* basada en Scrum y personalizada para adaptarse a las necesidades de este proyecto.

El software elegido para hacer la planificación y seguimiento de las tareas es *clickup [3]*, elegido por su facilidad de uso y versatilidad.

La planificación del trabajo se ha creado en base a las fechas de entrega de cada una de las PECs dividiendo las más largas en dos sprints, quedando una duración aproximada de 2 semanas por sprint.

En cada sprint se sigue una finalidad concreta y se ejecutan tareas en torno a ella:

- Sprint 0 (16-09 a 29-09): Planificación del Trabajo.

- Sprint 1 (30-09 a 14-10): Diseño del sistema y preparación del entorno de desarrollo.

- Sprint 2 (15-10 a 27-10): Desarrollo de la base del sistema y redacción de la memoria para la Entrega 2.

- Sprint 3 (28-10 a 11-11): Integración con TrustedX.

- Sprint 4 (12-11 a 24-11): Desarrollo del Sistema de Identidades y Autorización y redacción de la memoria para la Entrega 3.

- Sprint 5 (25-11 a 12-12): Integración de proveedores de almacenamiento en la nube (Dropbox y Google Drive).

- Sprint 6 (13-12 a 29-12): Últimos retoques y redacción de la memoria para la Entrega Final.

- Sprint 7 (30-12 a 05-01): Preparación y grabación de la presentación del Trabajo.

A continuación, se detallan los diagramas de Gantt de los distintos sprints con sus correspondientes tareas a ejecutar.

El sprint cero está enfocado a la organización y planificación del resto del trabajo, es donde se estudiarán las distintas APIs a integrar y la nueva regulación eIDAS.

Ilustración 1: Diagrama de Gantt - Sprint 'cero'

El primer sprint se centra en el diseño y la preparación del entorno de desarrollo, se pretende que después de este sprint las bases de la aplicación a desarrollar estén claras y poder minimizar las desviaciones posteriores que pueden hacer peligrar las fechas de entrega establecidas.

Ilustración 2: Diagrama de Gantt – Primer sprint

El segundo sprint tiene como objetivo conseguir una funcionalidad básica del servicio de firmas, creándolas de forma totalmente local y sin interfaz gráfica, el otro objetivo de este sprint es añadir en la memoria del trabajo los avances que se han hecho en los dos últimos sprints.

Ilustración 3: Diagrama de Gantt – Segundo sprint

En esta tercera fase se creará una primera versión de la interfaz gráfica y se integrará el servicio de firmas con los servidores de TrustedX.

Ilustración 4: Diagrama de Gantt – Tercer sprint

En la cuarta parte del trabajo se empieza a trabajar en la autorización de los recursos desde la aplicación, implementando una librería que ejecute el protocolo OAuth2 Code Grant.

Ilustración 5: Diagrama de Gantt – Cuarto sprint

La integración de los proveedores de documentos, Google Drive y Dropbox, serán los objetivos principales del siguiente sprint, también se pondrá foco en mejorar la interfaz gráfica.

Ilustración 6: Diagrama de Gantt – Quinto sprint

Las tareas de este sprint no se centran en nada en concreto, sino que se usarán para cerrar las últimas partes y hacer mejoras generales en el diseño y en funcionalidad. También será donde se redacte la parte final de la memoria.

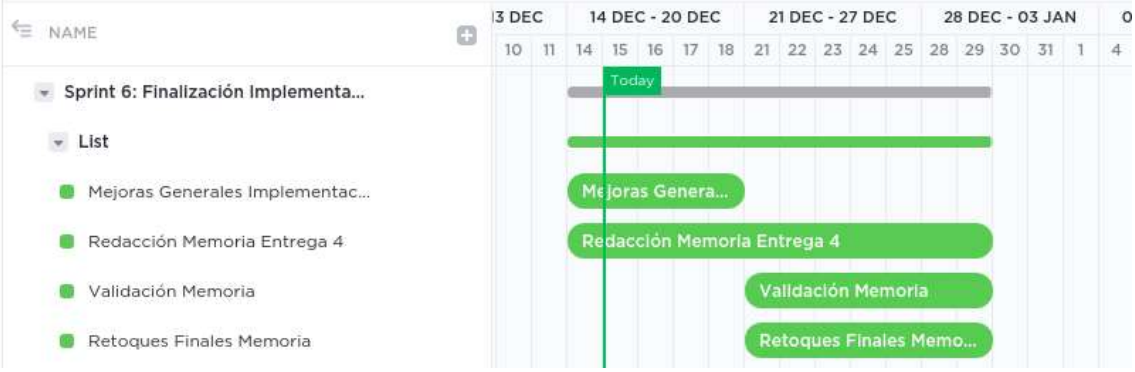


Ilustración 7: Diagrama de Gantt – Sexto sprint

En el último sprint, el desarrollo de la aplicación está totalmente terminado, así como la redacción de la memoria. Las tareas de este sprint son la preparación y grabación del video de la presentación del trabajo.

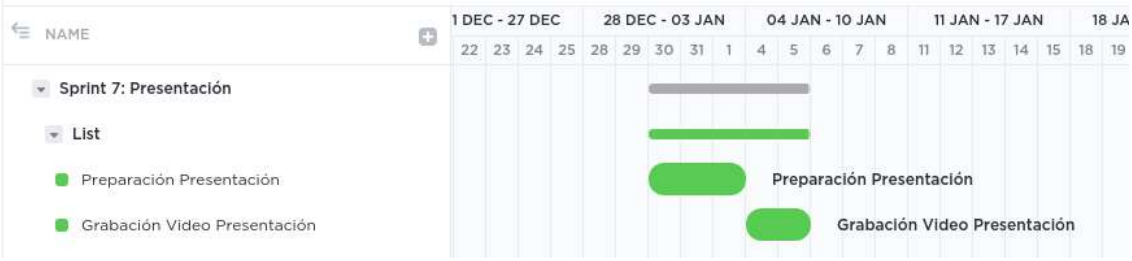


Ilustración 8: Diagrama de Gantt – Séptimo sprint

1.5 Breve sumario de productos obtenidos

Motivado por la nueva ley eIDAS que posibilita la firma electrónica cualificada de documentos en línea, se implementará un servicio de firma en la nube, con el que se podrán firmar documentos en línea, sin que haya ninguna diferencia a nivel legal con una firma física y que, además, no esté limitado por el soporte físico desde donde se esté haciendo la firma.

De forma más concreta, se habilitará la firma de PDFs almacenados en los proveedores de almacenamiento de documentos Dropbox y Google Drive, y se utilizará como proveedor de claves de firma a TrustedX.

De esta forma se explorarán las posibilidades y limitaciones que nos da la nueva regulación eIDAS.

1.6 Breve descripción de los otros capítulos de la memoria

En el próximo capítulo de la memoria se detalla el estado del arte actual y se irán introduciendo los detalles de la nueva normativa eIDAS y de la implementación de la aplicación, donde se hará mayor hincapié en el *framework* de autorización OAuth2 y en concreto con la autorización de tipo "*Code Grant*".

En el capítulo siguiente se exponen los detalles del diseño de la aplicación, se discutirán las distintas opciones que existen y el porqué de la elección de una opción sobre otra.

El tercer capítulo se trata la implementación de la aplicación, en este capítulo la aplicación ya estará finalizada y se darán detalles de las funcionalidades que provee y se citaran los casos de uso propuestos en el diseño que no se han podido implementar finalmente si los hubiera.

Por último, en el capítulo de conclusiones se hará un examen general del trabajo realizado, de los objetivos que se han conseguido y los que han faltado y se señalarán las posibles líneas futuras.

2. Estado del arte

En este apartado detallaremos el estado del arte tanto de la regulación de la firma electrónica digital como de los servicios web que ofrecen unas características parecidas al software desarrollado en este trabajo.

Para comprender los siguientes apartados es esencial saber los distintos tipos de firma que existen y que vinculación legal tienen:

- Firma electrónica.

El Reglamento eIDAS la define como “los datos en formato electrónico anejos a otros datos electrónicos o asociados de manera lógica con ellos que utiliza el firmante para firmar”. Prueba la aceptación del firmante mediante el uso de algún tipo de certificado o firma grafométrica.

Es el tipo de firma más básico, no pudiendo equipararse a la firma manuscrita.

- Firma electrónica avanzada.

Firma electrónica que además cumple que está vinculada al firmante de manera única y permite la identificación del firmante, ha sido creada con un alto nivel de confianza y bajo el control y uso exclusivo del firmante, y, por último, está vinculada y sellada con los datos firmados de modo tal que cualquier modificación posterior de los mismos sea detectable.

Es el tipo de firma que se encuentra en un nivel intermedio de seguridad. A pesar de permitir la vinculación con el firmante, y garantizar la certeza en cuanto al contenido del documento, no pudiendo equipararse a la firma manuscrita.

- Firma electrónica cualificada.

Firma electrónica avanzada que se crea mediante un dispositivo cualificado de creación de firmas electrónicas y que se basa en un certificado cualificado de firma electrónica.

Tiene el nivel de seguridad más alto y las máximas garantías jurídicas, equiparando a nivel legal a la firma manuscrita.

2.1 eIDAS

La regulación 910/2014 regula el mercado interior de servicios de confianza dentro de la Unión Europea. Esta regulación trata de ayudar en la implantación de un Mercado Digital Europeo Único [4].

eIDAS define los estándares para la firma electrónica simple, la firma electrónica avanzada y la firma electrónica cualificada, la emisión de certificados cualificados y los servicios de confianza online.

Esta nueva normativa fue completamente disruptiva en muchos sectores ya que introduce, por primera vez en Europa, métodos para la identificación electrónica, por lo que las entidades gubernamentales y las empresas no requieren la presencia física de la persona para realizar una operación que antes si la requería.

Gracias a eIDAS se ha podido digitalizar muchos procesos que reducen drásticamente la burocracia, reduciendo costes y tiempos a la vez que se facilita la experiencia de usuario. Para muchos procesos administrativos, antes se necesitaba el chip electrónico y un lector de tarjetas para conseguir la identificación digital, ahora simplemente con el móvil se podría realizar ese mismo proceso, siendo este método más seguro y además aumenta drásticamente la usabilidad.

El ETSI (Instituto Europeo de Normas Técnicas) tiene la función de emitir normas técnicas por delegación en el Reglamento eIDAS.

2.2 PaDeS

PAdES es un conjunto de normas técnicas, creadas por el ETSI, que tienen la finalidad de permitir la firma avanzada y cualificada de PDFs según dicta la normativa eIDAS. Con la nueva normativa, no se puede negar la validez de ninguna firma electrónica reconocida por eIDAS por ser electrónica [5].

PAdES tiene 4 niveles de verificación para certificado digital, desde el más simple hasta el más complejo que permite que los documentos firmados electrónicamente sigan siendo válidos durante períodos prolongados incluso si los algoritmos criptográficos subyacentes o los otros certificados caducados.

En este trabajo nos basaremos en este estándar para la firma de los documentos.

2.3 OAuth2

OAuth 2.0 permite que una aplicación de terceros obtenga acceso limitado a un servicio HTTP, ya sea en nombre del propietario de un recurso, o permitiendo que la aplicación obtenga acceso en su propio nombre [6].

En el modelo tradicional de autenticación cliente-servidor, el cliente solicita un recurso protegido en el servidor mediante la autenticación con el servidor utilizando las credenciales del propietario del recurso. Para esto, el propietario del recurso comparte sus credenciales con el tercero. Esto, evidentemente, crea varios problemas y limitaciones.

OAuth aborda estos problemas mediante la introducción de una capa de autorización y la separación de la función del cliente de la del propietario del recurso. En OAuth, el cliente solicita acceso a los recursos controlados por el propietario del recurso y alojados por el servidor de recursos, y se le emite un conjunto de credenciales diferente al del propietario del recurso.

El siguiente recurso representa el intercambio real de paquetes entre los diferentes agentes implicados en la autorización de un recurso *mediante OAuth2 Code Grant* en la aplicación desarrollada en este trabajo:

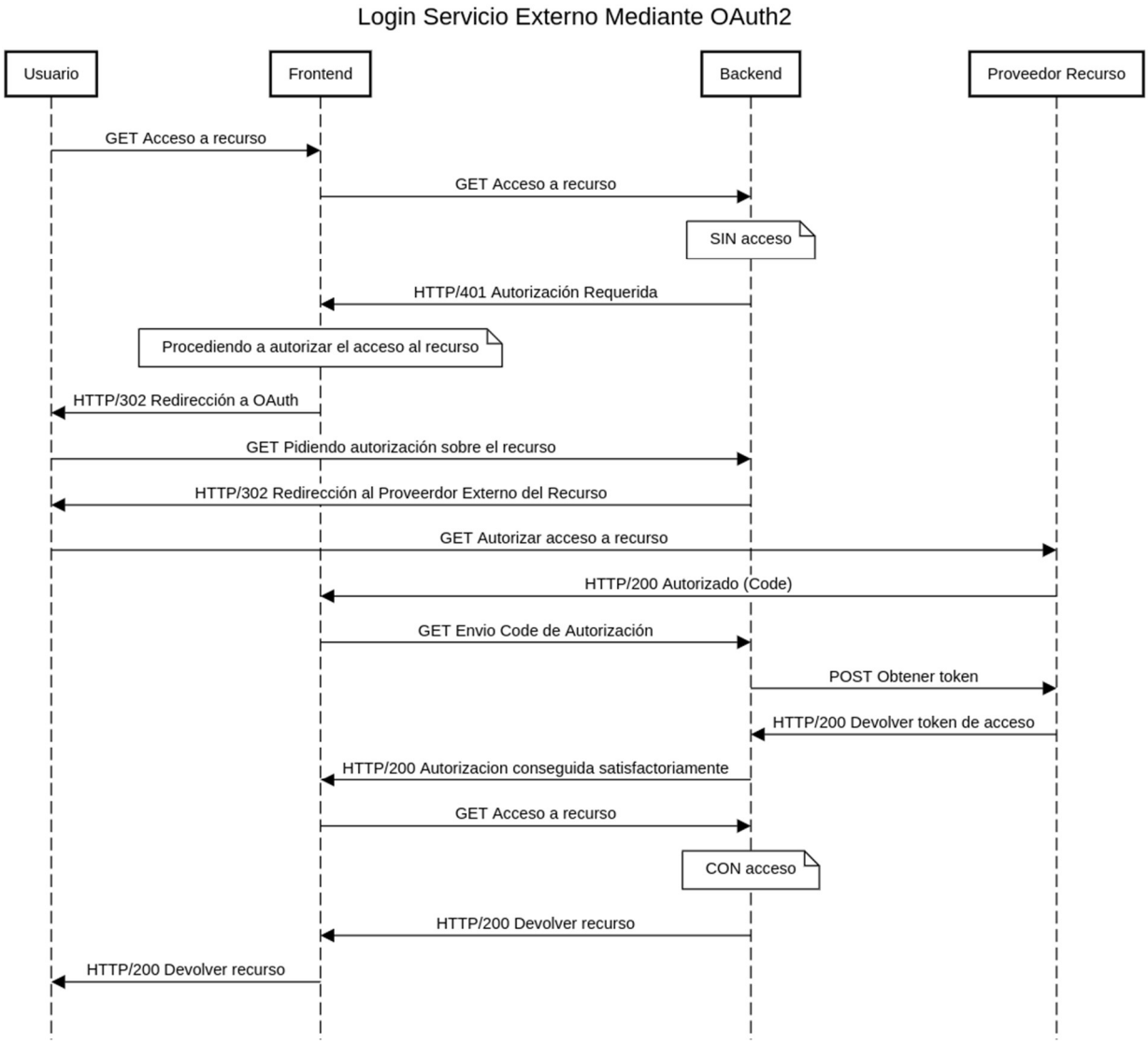


Ilustración 9: Login servicio externo OAuth2

En este trabajo se usará este protocolo para conseguir acceso a los distintos recursos del cliente (documentos PDF, claves de firma) que estarán alojados en los diferentes servicios de terceros (Google Drive, Dropbox, TrustedX).

2.4 Servicios de alojamiento de archivos

Un servicio de alojamiento de archivos es un servicio que permite a los usuarios alojar sus archivos en la nube.

Normalmente estos servicios vienen acompañados con una Interfaz Gráfica que permite cargar los archivos de forma sencilla y posteriormente la visualización de su contenido y una API que facilita la integración de estos proveedores con servicios externos.

2.4.1 Google Drive

Google Drive es el servicio de almacenamiento de archivos en la nube de Google. En 2012, Google Drive se convirtió en el sucesor de Google Docs y, junto a este reemplazo, se produjo una integración completa del almacenamiento de Drive con otros servicios de Google como Gmail.

Google Drive se ofrece gratuitamente a todos los usuarios que posean una cuenta de Google, lo que hace que tenga una gran base de usuarios que hace dos años (2018) ya superaban los mil millones [7].

En cuanto a las integraciones con terceros, que será lo que se usará principalmente en este trabajo, se realiza mediante Google Cloud, ofreciendo toda la potencia y posibilidades de esta.

La facilidad de integración de este servicio con terceros gracias a la integración con Google Cloud y la gran cantidad de usuarios que usan diariamente esta plataforma, son los motivos principales por los que se ha escogido esta plataforma para el trabajo.

2.4.2 Dropbox

Dropbox es un servicio de alojamiento de archivos de la empresa estadounidense Dropbox Inc., con sede en San Francisco. Fue fundada en 2007 por los estudiantes del MIT, Drew Houston y Arash Ferdowsi.

Dropbox ofrece almacenamiento en la nube y sincronización de archivos. Actualmente la plataforma cuenta con más de 600 millones de usuarios, de los cuales alrededor de 15 millones son cuentas premium [8]. Esto convierte a Dropbox en una de las plataformas de almacenamiento de archivos más usadas en el mundo.

Una característica de este servicio es el foco que existe en hacer el servicio usable por terceras aplicaciones gracias a su API REST y a la gran

cantidad de clientes oficiales que existen en los lenguajes de programación más usados.

Al igual que Google Drive, se ha escogido esta plataforma por la gran base de usuarios que tiene y la facilidad que provee a la hora de integrar aplicaciones de terceros.

2.5 Servicio de custodia de claves y certificados digitales

La custodia de claves y certificados digitales permite almacenar y gestionar información privada de personas y entidades en la nube, aumentando la seguridad de estas respecto a la gestión de forma local ya que se disponen de mecanismos certificados de control y protección, y hardware especializado para el almacenamiento de claves.

El nuevo reglamento eIDAS hace posible una solución de firma electrónica cualificada sin necesidad de tarjetas criptográficas. Las soluciones de firma electrónica con custodia remota de claves presentan numerosas ventajas para el usuario respecto a las aproximaciones más tradicionales.

Un modelo basado en claves centralizadas libera al usuario de la necesidad de disponer de un lector de tarjetas y de los problemas de configuración que estos conllevan. También le permite utilizar su firma electrónica desde dispositivos móviles.

Las claves privadas de los usuarios son custodiadas en un repositorio único, facilitando su gestión y la seguridad. Además, estas se almacenan en un módulo criptográfico (HSM) desde donde se realizan todas las operaciones criptográficas, de manera que las claves nunca salen de este entorno protegido.

Otra de las ventajas que ofrecen los sistemas de custodia es que garantizan que las claves solo puedan ser utilizadas por su propietario mediante diferentes controles de acceso que cumplen con la regulación actual. Esta regulación obliga a los servicios a implementar una autenticación fuerte, en la que, además de una autenticación típica de usuario y contraseña, se exige a los usuarios un segundo factor de autenticación, como puede ser un código de un solo uso (OTP).

Al estar estos servicios normalmente en la nube, podemos disfrutar todas las ventajas que la *cloud* ofrece, como por ejemplo poder usar estas claves y certificados desde cualquier sitio y dispositivo, con una alta disponibilidad del servicio.

Además de estas ventajas, los servicios de custodia de claves suelen incluir funcionalidades que aumentan la capacidad de gestión y la seguridad de las claves. Un ejemplo de estas funcionalidades adicionales que añaden los proveedores es la monitorización de las claves y del registro del uso de estas.

Estos servicios han vivido en estos últimos años un gran aumento de su uso impulsado por la digitalización de los procesos dentro de las empresas y organismos gubernamentales y el aumento del teletrabajo.

2.5.1 TrustedX

TrustedX es el servicio de custodia de claves y certificados digitales de la empresa española Safelayer.

Según su propia web, TrustedX es "una plataforma de servicios web para la integración de mecanismos de identificación, autenticación y firma electrónica (eIDAS) de usuarios. Además de proporcionar un conjunto de funciones de autenticación multi-factor, gestión del nivel de la confianza y federación de identidades, proporciona firma remota con claves PKI en servidor a través de API Web" [9].

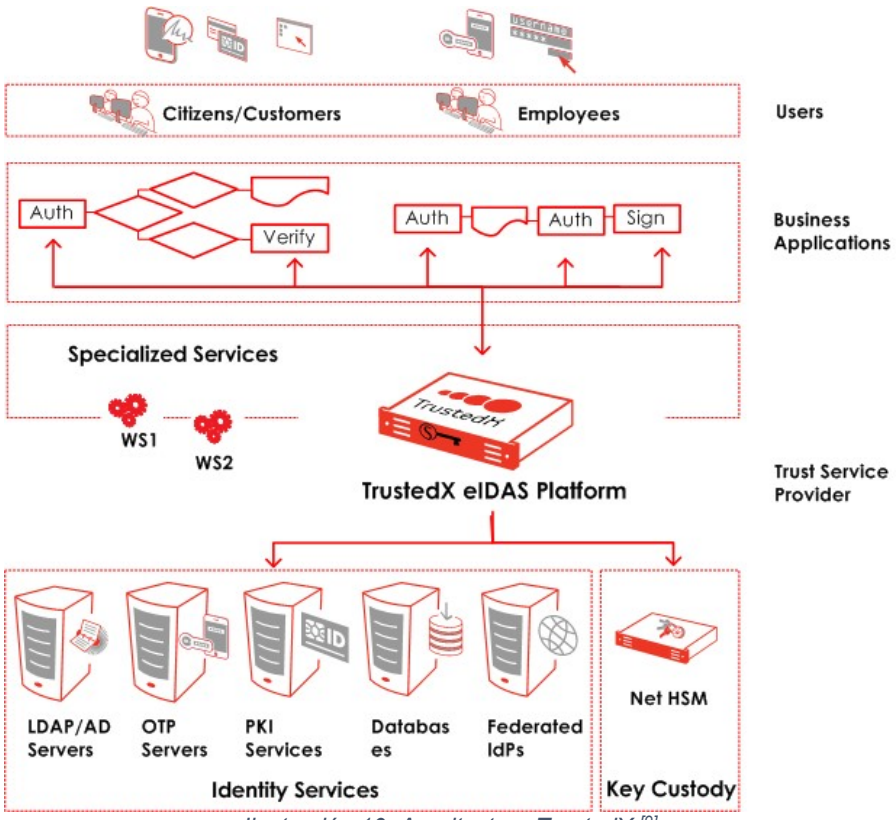


Ilustración 10: Arquitectura TrustedX [9]

TrustedX nos proporciona una solución completa diseñada para proporcionar identificación segura de usuarios en un contexto de movilidad y nube, posee funciones de firma remota según los estándares técnicos CEN TS 419 241 y ofrece una API REST que facilita la integración con aplicaciones de terceros. En este trabajo utilizaremos todas estas utilidades que nos ofrece la plataforma TrustedX.

2.6 Servicios Similares

Existen varios servicios web que ofrecen utilidades similares a las que vamos a implementar en este trabajo, el más extendido es docuSign.

La empresa docuSign tiene su sede en Estados Unidos y permite gestionar acuerdos y firmas digitales en distintos dispositivos electrónicos. Cumplen tanto con la regulación de EE. UU., ESIGN, como con la europea eIDAS [10].

Entre las funcionalidades que ofrecen, permiten la firma de documentos PDF en la nube y tienen integraciones con los principales proveedores de servicios de almacenamiento de archivos online.

3. Diseño

3.1 Arquitectura a alto nivel

Antes de proceder con el diseño de la aplicación, se deben detallar los casos de uso que esta debe cubrir, de esta forma se puede elegir objetivamente la arquitectura que mejor se adapte a lo requerido en estos casos de uso:

A. Autorización de gestión de ficheros en Google Drive y Dropbox.
B. Gestión mínima de documentos almacenado en Google Drive y Dropbox que por lo menos debe incluir las siguientes operaciones:
    a. Listar documentos PDF.
    b. Descarga de documentos PDF.
    c. Eliminación de documentos PDF.
    d. Subida de nuevos ficheros PDF.
C. Autorización de gestión de identidades de firma en TrustedX.
D. Gestión mínima de identidades de firma en TrustedX que por lo menos debe incluir las siguientes operaciones:
    a. Listar identidades.
    b. Detallar información de una identidad.
    c. Eliminación de identidades de firma.
    d. Creación de nuevas identidades de firma.
E. Autorización de uso de una identidad de firma para la firma de un documento.
F. Posibilidad de firmar digitalmente un documento PDF almacenado en Google Drive o Dropbox con una clave de firma almacenada en TrustedX.

Teniendo en cuenta todos estos casos de uso, se ha optado por una aplicación web para la implementación del servicio de firmas por la versatilidad que nos ofrece, pudiendo utilizar tanto en ordenadores como en dispositivos móviles, y por la velocidad del desarrollo.

Esta aplicación web estará montada sobre una arquitectura de microservicios, lo que resultará en una aplicación mucho más sólida y resiliente y, además, nos dará mayor versatilidad a la hora de escoger los lenguajes de programación, para cada característica a implementar podremos escoger el lenguaje más apropiado.

A muy alto nivel el diseño de la aplicación quedaría formado por un cliente que realiza las peticiones, una API que expone una interfaz hacia todos los servicios que se ofrecen, y el ‘frontend’ que expone una interfaz gráfica que facilita al cliente el uso de tales servicios:

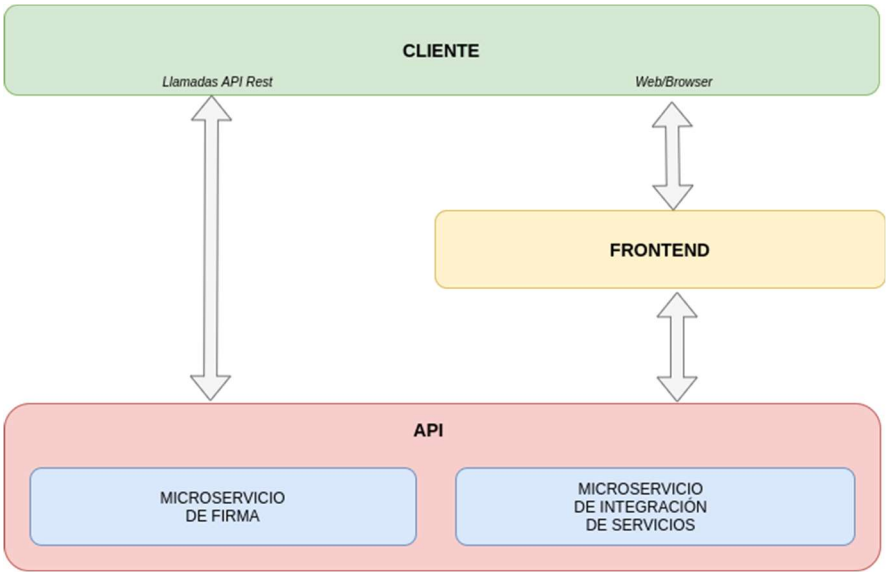


Ilustración 11: Diagrama Global Alto Nivel

El *backend* (marcado de color rojo en el diagrama anterior) estará compuesto por dos microservicios que formarán una *API REST* con la que el cliente podrá comunicarse, tanto directamente como indirectamente mediante la interfaz web expuesta por el frontend.

Los distintos componentes son totalmente independientes, lo que facilita su contenerización.

Los lenguajes de programación escogidos son *NodeJS* para el microservicio de integración de servicios, que será el encargado de crear una interfaz con todos los servicios externos que se utilizarán (TrustedX, Google Drive, Dropbox) y Java para el servicio de firma.

Enfocándonos en el backend, el diagrama quedaría tal que:

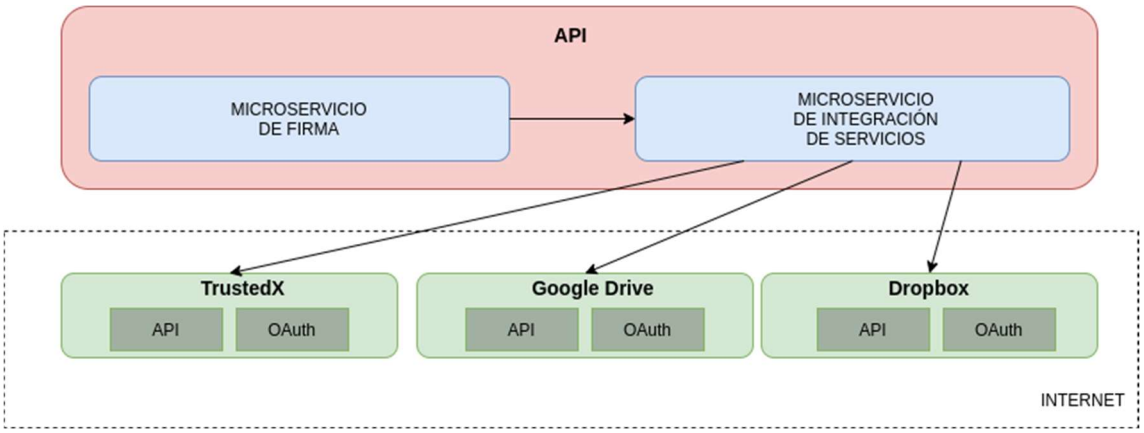


Ilustración 12: Diagrama Backend Alto Nivel

El microservicio de integración de servicios incluye la mayor parte de la funcionalidad, se ha decidido implementar en *NodeJS* por la gran cantidad de librerías que facilitan el desarrollo de una *API REST* y la integración con otras

APIs y con servicios OAuth2. En concreto se ha utilizado la librería `*express*` para crear las distintas rutas de la API y la librería `*simple-oauth*` para integrar los distintos proveedores OAuth2.

Por otro lado, el microservicio de firma se ha decidido crear en *Java* ya que este lenguaje ofrece una librería de gestión de PDFs que nos permitirá sacar el mayor provecho de los distintos componentes.

Un aspecto importante que se puede deducir del diagrama anterior es que la aplicación no incorpora ningún componente que gestione la autenticación y autorización del usuario. Se ha decidido delegar la autorización de los recursos a los propios servicios externos mediante OAuth2, y las credenciales de acceso a estos servicios externos estarán guardadas temporalmente en la sesión del usuario (más adelante se detallará este proceso). De esta forma, no se guarda información de los usuarios, lo que evita tener que lidiar con las leyes de protección de datos, y a la vez, facilita el diseño y la implementación, debido a que no es necesaria una base de datos que guarde esta información.

Una decisión por tomar en este componente es el método por el que se firmarán los documentos. El servicio TrustedX permite dos formas de firmado, la primera sería mediante la generación de una firma PKCS#1 sobre el resumen de los datos (hash) a firmar según el estándar de firma, este mecanismo permite no enviar el documento PDF al servidor TrustedX, con las ventajas que esto conlleva.

El segundo método sería subiendo el documento PDF al servicio de TrustedX y realizar la firma PAdES completamente en el servidor, de esta forma, no hace falta procesar el PDF con ninguna herramienta que genere la firma PAdES ya que se encargaría TrustedX.

Se ha escogido la primera opción ya que, aunque nos obligue a implementar el procesado del documento, minimizamos el consumo de ancho de banda y, sobre todo, aumentamos la privacidad del documento, ya que no abandona el servicio en ningún momento.

El *frontend* está basado en el *framework Angular*, esta elección se debe al enfoque que posee este *framework* que permite implementar un modelo Vista-Controlador (MVC) con mucha facilidad. Esto obliga a seguir un modelo de trabajo definido, mejorando el orden del proyecto y forzando a que se sigan buenas prácticas.

Por último, el diseño del frontend se construirá a partir del *framework* de diseño *Material*, este módulo incluye varios componentes de angular que permiten crear una interfaz muy agradable de forma sencilla, lo que encaja perfectamente en los objetivos de este trabajo.

3.2 Arquitectura a bajo nivel

Una vez detallada la arquitectura a alto nivel, se puede empezar a concretar detalles del desarrollo.

En las siguientes secciones se explicará con detalle la arquitectura de la API REST y de la interfaz gráfica.

3.2.1 API REST

Como se ha comentado anteriormente, el backend está dividido en dos microservicios diferentes, aunque los dos microservicios se han diseñado siguiendo una arquitectura de *API REST* para facilitar su interpretación e integración con el resto de los componentes del diseño y posibles servicios externos futuros.

La arquitectura de *API REST* nació con la intención de simplificar el desarrollo y el consumo de las APIs tradicionales basadas en *CORBA*, *RPC* o *SOAP*, al contrario que estas, una *API REST* se basa directamente en *HTTP*.

Empezando con el microservicio de integración de servicios externos, podemos encontrar dos tipos llamadas posibles, las de autorización, bajo la dirección *"/oauth"* y la de operaciones sobre recursos, bajo la dirección *"/api/v1"*.

El motivo por el que se ha decidido no incluirlos también bajo la dirección *"/api/v1"* aun estando implementados junto al resto de llamadas es que, al estar basadas en autorizaciones *OAuth2 Code Grant*, es necesario el uso de un navegador para poder realizar dicha autorización y esto entra en conflicto con la propia definición de una *API REST* que expone que debe ser independiente del cliente desde el que se realizan las peticiones.

La forma en la que se gestionarán los tokens de autorización es mediante sesiones de usuario basadas en cookies, el microservicio que implementará esta funcionalidad es el de integración con servicios externos, ya que es este el que establece la interfaz con los servicios de terceros.

Al usuario la primera vez que ejecuta una petición a la API, se le asigna una cookie de sesión que deberá ser enviada junto con todas las peticiones posteriores para poder identificar qué se trata de la misma sesión.

En la parte del servidor, existe una base de datos que contiene una entrada por cada una de las sesiones activas que existen en ese momento y que es compartida por todas las aplicaciones que necesitan esta información, en nuestro caso estos datos no se almacenan en una base de datos, sino que quedan almacenados en la memoria del microservicio.

A cada entrada de esta base de datos de sesiones se le pueden incluir datos específicos de esa sesión, en nuestro caso incluimos los tokens de acceso a los servicios externos que el usuario ha concedido.

A estas sesiones se les asigna un tiempo de vida y son destruidas si el cliente no envía nuevas peticiones antes de ese tiempo.

A continuación, se muestra un ejemplo de cómo funciona este mecanismo en nuestro caso:

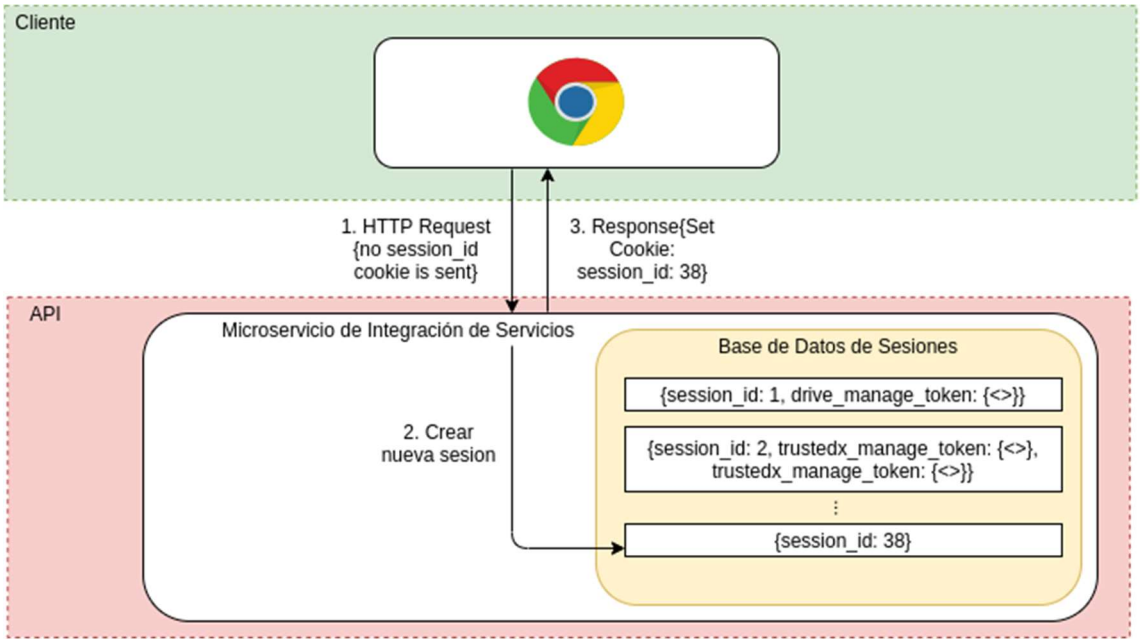


Ilustración 13: Ejemplo de uso de sesión

En el diagrama se muestra cómo el usuario con el número de sesión 1 ha autorizado al microservicio a acceder a su cuenta de Google Drive con permisos de gestión, y el usuario 2 ha autorizado el acceso a TrustedX y a Google Drive (lo que incluye estos permisos se detalla más adelante).

Por otro lado, el usuario del ejemplo o, no ha accedido con anterioridad a la aplicación, o su sesión anterior ha caducado, por este motivo el microservicio responde creando una nueva entrada en la base de datos de sesiones y haciendo un *"Set Cookie"* con el identificador de la sesión que se acaba de crear.

Si este nuevo usuario autoriza a la aplicación el acceso a un servicio externo, estas credenciales de acceso se guardarán en la base de datos de sesiones en la entrada que corresponde a ese usuario.

3.2.2 Interfaz Gráfica

La interfaz gráfica debe contener al menos seis vistas independientes para poder ejecutar todos los casos de uso expuestos al comienzo de este apartado, de las cuales cuatro vistas serían principales y las otras dos serían vistas derivadas a partir de otras.

La primera vista detalla las funcionalidades de la aplicación, permitiendo al usuario conocer rápidamente cuales son las funcionalidades que ofrece. Otra vista sería necesaria para la gestión de documentos PDF, permitiendo ejecutar todas las operaciones descritas. De igual modo, la gestión de identidades de firma también necesita una vista propia. Por último, la última vista principal sería la vista de firma, donde se permite que se firme un documento PDF de un proveedor de archivos externo con una clave de firma almacenada en TrustedX.

Las dos sub-vistas serían los formularios que permiten la subida de nuevos documentos PDFs y nuevas identidades de firma, ya que incluir estos formularios en las vistas de gestión de sus respectivos recursos haría que estas vistas quedarán poco manejables.

A continuación, se incluye un diagrama de las vistas incluidas y como sería el flujo entre ellas:

Ilustración 14: Modelo de Vistas

4. Implementación

En esta sección se discutirán los detalles de la implementación. Se ha dividido en 3 partes, en la primera se discutirá sobre la implementación de la API, en la segunda hablaremos sobre la Interfaz Gráfica y, por último, veremos cuál ha sido el método de despliegue y la infraestructura donde se ha hecho.

4.1 API

Tal y como se ha comentado en la sección anterior, la autorización de los recursos por parte del usuario se ejecutará mediante OAuth2 y se guardarán las credenciales en una cookie de sesión. Esta cookie de sesión se identifica con *'connect.id'*, este es el nombre por defecto que utiliza la librería *`express-session'* y no se ha visto necesario cambiarlo.

A continuación, se detallan cada una de las llamadas a la API empezando por las que tienen relación con la autorización sobre los recursos.

4.1.1 Autorización de gestión sobre Google Drive

a) Ubicación:
   Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
   Autoriza a `SignPDF` la gestión de archivos alojados en tu cuenta de Google Drive.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - [Opcional] "redirect_uri": URL donde redireccionar la petición una vez autenticada.
   ii) Body:
      - Nada
   iii) Headers:
      - Nada

e) Detalles de implementación:
   Redirecciona al servicio de OAuth2 de Google Drive para pedir autorización sobre el scope "https://www.googleapis.com/auth/drive.file".

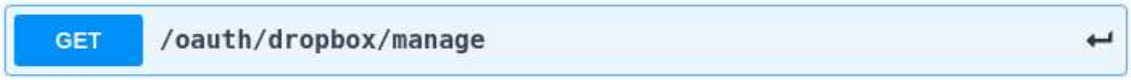
4.1.2 Autorización de gestión sobre ficheros almacenadas en Dropbox

a) Ubicación:

Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Autoriza a `SignPDF` la gestión de archivos alojados en tu cuenta de Dropbox.

c) Petición:



d) Parámetros de entrada:
   i) Query:
      - [Opcional] “redirect_uri”: URL donde redireccionar la petición una vez autenticada.
   ii) Body:
      - Nada
   iii) Headers:
      - Nada

e) Detalles de implementación:
Redirecciona al servicio de OAuth2 de Dropbox para pedir autorización sobre los scopes "file_requests.read”, “file_requests.write”, “files.content.read” y “files.content.write".

4.1.3 Autorización de gestión sobre TrustedX

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Autoriza a `SignPDF` la gestión de identidades de firma alojadas en tu cuenta de TrustedX.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - [Opcional] “redirect_uri”: URL donde redireccionar la petición una vez autenticada.
   ii) Body:
      - Nada
   iii) Headers:
      - Nada

e) Detalles de implementación:

Redirecciona al servicio de OAuth2 de TrustedX para pedir autorización sobre los scopes "urn:safelayer:eidas:sign:identity:register" y "urn:safelayer:eidas:sign:identity:manage".

4.1.4 Autorización de uso de una identidad de firma

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Autoriza a SignPDF` la utilización de la identidad de firma seleccionada para la firma.

c) Petición:

d) Parámetros de entrada:
i) Query:
- [Opcional] "redirect_uri": URL donde redireccionar la petición una vez autenticada.

ii) Body:
- Nada

iii) Headers:
- Nada

e) Detalles de implementación:
Redirecciona al servicio de OAuth2 de Google Drive para pedir autorización sobre el scope "https://www.googleapis.com/auth/drive.file".

4.1.5 Listado de las autorizaciones concedidas en la sesión

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Lista todas las autorizaciones concedidas a SignPDF en la sesión actual.

c) Petición:

d) Parámetros de entrada:
i) Query:
- Nada

ii) Body:
- Nada

iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Detalles de implementación:
Devuelve el contenido de la base de datos de sesiones tras filtrar los campos con credenciales.

Una vez descritas las peticiones de autorización, pasamos a detallar las peticiones que ejecutan una operación sobre un recurso.

Como veremos a continuación, se ha intentado crear una interfaz homogénea entre las peticiones que se dirigen a servicios de Google Drive y las que están destinadas a Dropbox, de esta manera se consigue facilitar la integración del frontend y de futuras posibles terceras aplicaciones.

4.1.6 Listar documentos PDF almacenados en Google Drive

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Lista todos los documentos PDF almacenados en la cuenta Google Drive a la que se le ha autorizado el acceso previamente. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

d) Parámetros de entrada:
i) Query:
- Nada
ii) Body:
- Nada
iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión archivos Google Drive

f) Detalles de implementación:
Se realiza una petición GET a la API de Google Drive https://www.googleapis.com/drive/v3/files incluyendo un filtro para que se devuelvan solo los documentos con mimeType igual a 'application/pdf'.

4.1.7 Listar documentos PDF almacenados en Dropbox

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Lista todos los documentos PDF almacenados en la cuenta Dropbox a la que se le ha autorizado el acceso previamente. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

```
GET  /api/v1/dropbox/pdf  ↵
```

d) Parámetros de entrada:
i) Query:
- Nada

ii) Body:
- Nada

iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión archivos Dropbox

f) Detalles de implementación:
Se realiza una petición POST a la API de Dropbox https://api.dropboxapi.com/2/files/search_v2 incluyendo la opción *file_categories: ["pdf"]*.

4.1.8 Listar identidades de firma almacenadas en TrustedX

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Lista todas las identidades de firma almacenadas en la cuenta TrustedX a la que se le ha autorizado el acceso previamente. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

```
GET  /api/v1/trustedx/signidentity  ↵
```

d) Parámetros de entrada:
i) Query:
- Nada

ii) Body:
- Nada

iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión claves de firma de TrustedX

f) Detalles de implementación:
Se realiza una petición GET a la API de TrustedX https://uoc.safelayer.com:8082/trustedx-resources/esigp/v1/sign_identities.

4.1.9 Descarga de documento PDF almacenado en Google Drive

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Descargar el archivo PDF seleccionado. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

d) Parámetros de entrada:
i) Query:
- Nada
ii) Body:
- Nada
iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión archivos Google Drive

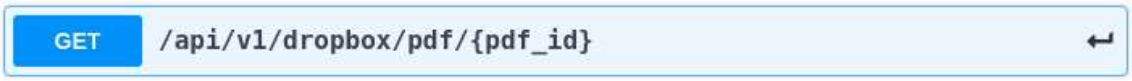
f) Detalles de implementación:
Se realiza una petición GET a la API de Google Drive https://www.googleapis.com/drive/v3/files/${fileId}.

4.1.10 Descarga de documento PDF almacenado en Dropbox

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Descargar el archivo PDF seleccionado. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:



d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - Nada
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Dropbox

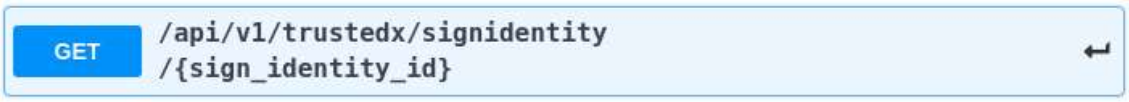
f) Detalles de implementación:
   Se realiza una petición POST a la API de Dropbox https://content.dropboxapi.com/2/files/download incluyendo el header con los parámetros *"Dropbox-API-Arg": `{"path": "${fileId}"}`*.

4.1.11 Detallar identidad de firma almacenada en TrustedX

a) Ubicación:
   Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
   Devuelve la información detallada de la identidad de firma seleccionada.

c) Petición:



d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - Nada
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Google Drive

f) Detalles de implementación:
   Se realiza una petición GET a la API de TrustedX https://uoc.safelayer.com:8082/trustedx-resources/esigp/v1/sign_identities/${sign_identity_id}

4.1.12 Borrado de documento PDF almacenado en Google Drive

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Elimina el documento PDF que se ha seleccionado de Google Drive. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

```
DELETE /api/v1/drive/pdf/{pdf_id} ↵
```

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - Nada
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Google Drive

f) Detalles de implementación:
Se realiza una petición DELETE a la API de Google Drive https://www.googleapis.com/drive/v3/files/${fileId}

4.1.13 Borrado de documento PDF almacenado en Dropbox

a) Ubicación:
Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
Elimina el documento PDF que se ha seleccionado de Dropbox. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

```
DELETE /api/v1/dropbox/pdf/{pdf_id} ↵
```

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - Nada
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Dropbox

f) Detalles de implementación:
   Se realiza una petición POST a la API de Dropbox https://api.dropboxapi.com/2/files/delete_v2 mandando en el body un json con el ID del documento a borrar.

4.1.14 Borrado de identidad de firma almacenada en TrustedX

a) Ubicación:
   Microservicio de integración de servicios externos (NodeJs).

b) Descripción:
   Elimina la identidad de firma que se ha seleccionado de TrustedX. Devuelve el código HTTP 403 si no se ha concedido acceso a la cuenta.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - Nada
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión identidades de firma de TrustedX

f) Detalles de implementación:
   Se realiza una petición DELETE a la API de TrustedX https://uoc.safelayer.com:8082/trustedx-resources/esigp/v1/sign_identities/${sign_identity_id}`

4.1.15 Subida de nuevo documento PDF a Google Drive

a) Ubicación:
   Microservicio de integración de servicios externos (NodeJs).

b) Descripción:

Sube un archivo PDF local a la cuenta de Google Drive que ha sido previamente autorizada. El documento se sube al directorio raíz y no se puede especificar otro directorio alternativo.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - [Obligatorio] form-data con el archivo PDF con la clave *'file'*
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Google Drive

f) Detalles de implementación:

Esta petición se ejecuta en dos partes, en la primera se recibe el archivo y se guarda en una carpeta temporalmente, una vez está el archivo completamente subido, se hace una petición de creación de un fichero a la API de Google Drive utilizando el SDK que ofrece.

4.1.16 Subida de nuevo documento PDF a Dropbox

a) Ubicación:

Microservicio de integración de servicios externos (NodeJs).

b) Descripción:

Sube un archivo PDF local a la cuenta de Dropbox que ha sido previamente autorizada. El documento se sube al directorio raíz y no se puede especificar otro directorio alternativo.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body:
      - [Obligatorio] form-data con el archivo PDF con la clave *'file'*
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión archivos Dropbox

f) Detalles de implementación:

Esta petición se ejecuta en dos partes, en la primera se recibe el archivo y se guarda en una carpeta temporalmente, una vez está el archivo completamente subido, se hace una petición de creación de un fichero a la API de Dropbox utilizando el SDK que ofrece.

4.1.17 Creación de nueva identidad de firma en TrustedX

a) Ubicación:

Microservicio de integración de servicios externos (NodeJs).

b) Descripción:

Crea una nueva identidad de firma en TrustedX

c) Petición:

d) Parámetros de entrada:

i) Query:
- Nada

ii) Body (JSON):
- [Obligatorio] labels: array de etiquetas
- [Obligatorio] pkcs12: certificado PKCS12 en base64
- [Obligatorio] password: contraseña de certificado

iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión identidades de firma de TrustedX

f) Detalles de implementación:

Se realiza una petición GET a la API de TrustedX con los datos obtenidos en el body de la petición https://uoc.safelayer.com:8082/trustedx-resources/esigp/v1/sign_identities/server/pki_x509/pkcs12

4.1.18 Firma de hash con una identidad de firma de TrustedX

a) Ubicación:

Microservicio de integración de servicios externos (NodeJs).

b) Descripción:

La petición devuelve el hash enviado firmado con la clave de firma especificada con el algoritmo SHA256.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body (JSON):
      - [Obligatorio] digest_value: hash a firmar
      - [Obligatorio] sign_identity_id: identificador de la identidad de firma con la que se quiere firmar el hash
   iii) Headers:
      - [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
   - Gestión identidades de firma de TrustedX

f) Detalles de implementación:
   Se realiza una petición POST a la API de TrustedX https://uoc.safelayer.com:8082/trustedx-resources/esigp/v1/signatures/server/raw pasando en el body los datos que ha mandado el usuario, más el algoritmo usado para la firma (*signature*_algorithm) que siempre es rsa-sha256.

4.1.19 Firma de un documento PDF

a) Ubicación:
   Microservicio de firmas (Java).

b) Descripción:
   Esta llamada firma un documento PDF almacenado en Google Drive o Dropbox con una firma almacenada en TrustedX. Se devuelve el PDF firmado.

c) Petición:

d) Parámetros de entrada:
   i) Query:
      - Nada
   ii) Body (JSON):
      - [Obligatorio] pdf_id: identificador del PDF a firmar
      - [Obligatorio] sign_identity_id: identidad de firma con la que se pretende firmar el documento.
      - [Obligatorio] reason: razón a poner en la firma
      - [Obligatorio] location: localización a poner en la firma
      - [Obligatorio] provider: proveedor del documento PDF a firmar, seleccionar “drive” para Google drive o “dropbox” para Dropbox

iii) Headers:
- [Obligatorio] Cookie de sesión (*connect.sid="..."*)

e) Permisos necesarios:
- Gestión archivos Google Drive o Dropbox (según el valor del 'provider')
- Autorización de uso de clave de firma seleccionada
- Gestión de identidades de firma de TrustedX

f) Detalles de implementación:

Esta es la petición más compleja, ya que es la única que ejecuta internamente llamadas a la propia API. Una vez recibida la petición, el microservicio procede a descargar el certificado x509 de la identidad de firma elegida y el PDF especificado del proveedor de documentos seleccionado en el campo '*provider*'. Una vez que tiene el PDF, procede a calcular el hash SHA256 de los datos a firmar y procede a pedir la firma del hash, una vez que tiene la firma y el certificado, lo introduce en el PDF para firmarlo y lo devuelve al usuario.

4.2 Interfaz Gráfica

Toda la interfaz gráfica está basada en la librería '*Material*', gracias a esto se consigue un diseño '*Responsive*', haciéndola usable tanto en ordenadores como en dispositivos móviles y tablets.

Se ha creado una interfaz gráfica simple pero que incluye todos los casos de uso citados en apartados anteriores. La interfaz gráfica se puede dividir en dos partes, las vistas implementadas y el menú principal que se incluye en todas las vistas, este menú tiene el propósito de facilitar la navegación por la página e incluir datos importantes para el usuario.

4.2.1 Menú principal

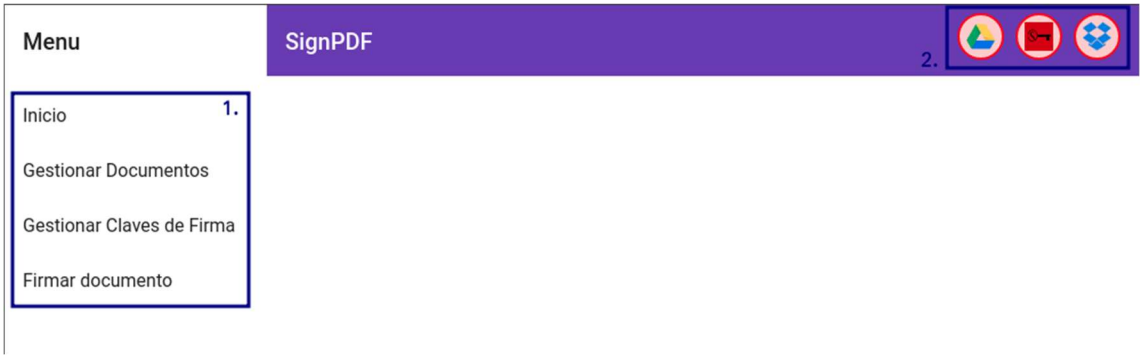


Ilustración 15: Menú Principal

El menú principal contiene dos componentes, el primero es el menú de navegación donde se pueden ver las distintas vistas principales que existen. El segundo componente informa si se ha autorizado o no a los distintos

componentes, el fondo es verde si ‘SignPDF’ está autorizado a acceder a ese servicio en la sesión actual y rojo si no está autorizado.

4.2.2 Página Inicio

Ilustración 16: Página Inicio

La vista de la página de inicio es la más simple ya que solo contiene un texto explicativo de las posibilidades que ofrece la página.

4.2.3 Gestión de documentos

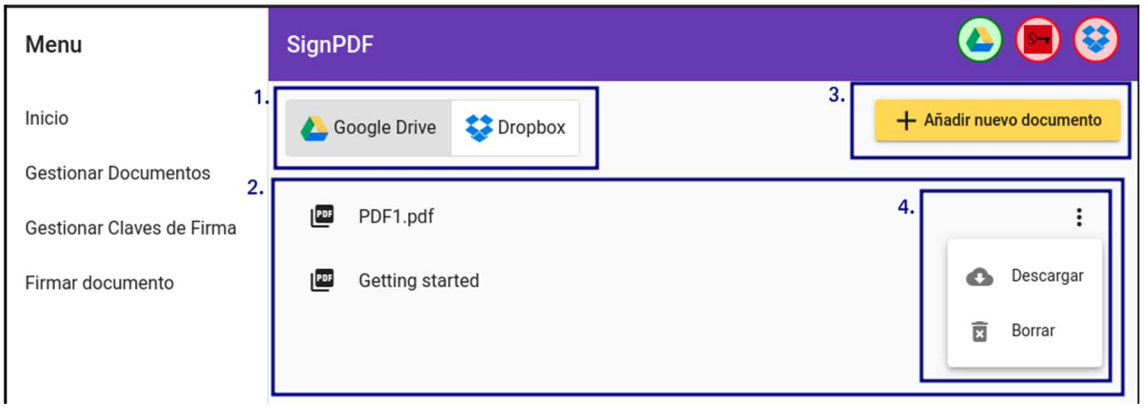


Ilustración 17: Página Gestión Documentos

Esta vista permite la gestión de los documentos PDF, el componente 1 permite seleccionar el proveedor de documentos, el segundo componente es la lista de PDFs que hay en la cuenta seleccionada. El componente 3 es un botón que redirige al usuario a la vista de subida de nuevos documentos, por último, el componente 4 es un menú que incluye cada documento con las distintas operaciones que se pueden ejecutar contra el.

4.2.4 Subida de documentos

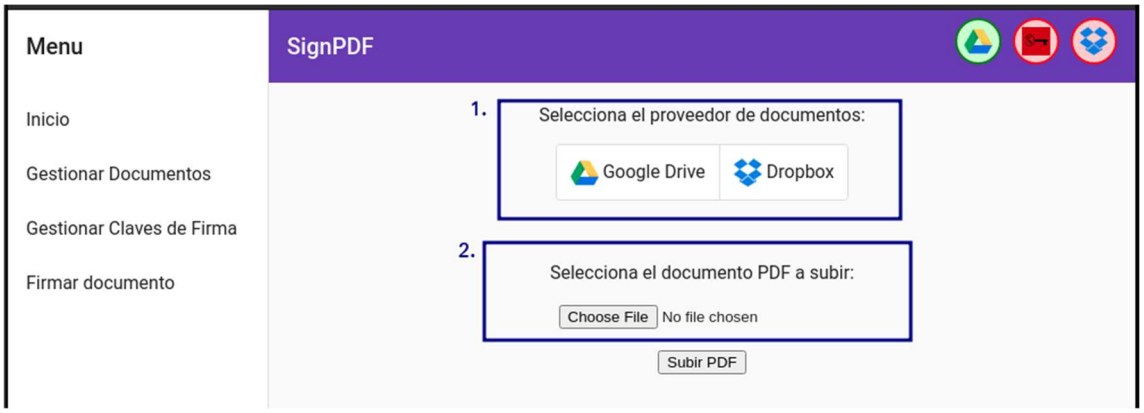


Ilustración 18: Página Subida Documentos

Esta vista contiene dos componentes, el primero es para seleccionar a qué proveedor de documentos se va a subir el PDF y el segundo es para escoger el PDF guardado en local que se quiere subir.

4.2.5 Gestión de identidades de firma

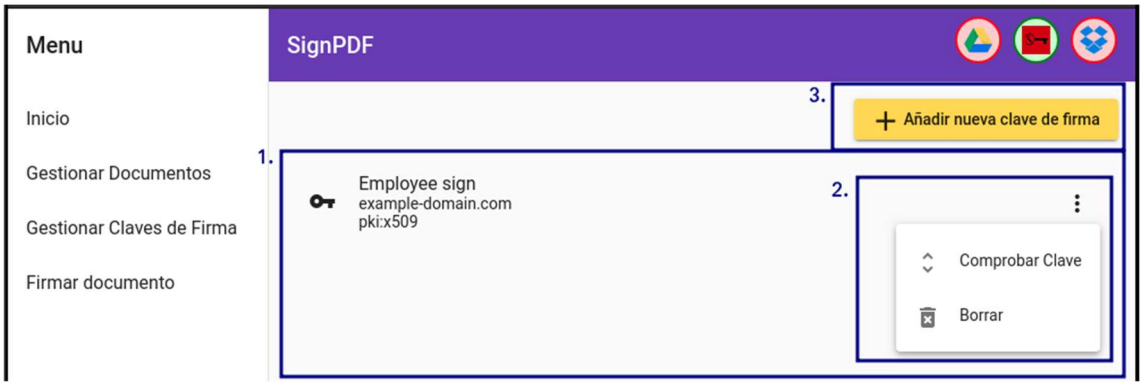


Ilustración 19: Página Gestión Identidades de Firma

Esta vista permite la gestión de las identidades de firma. El primer componente es la lista de identidades de firma que hay en la cuenta de TrustedX del usuario. El segundo componente es el menú que incluye cada una de las identidades de firma listadas con las distintas operaciones que se pueden ejecutar.

Por último, el componente 3 es un botón que redirige al usuario a la vista de subida de nuevas identidades de firma.

4.2.6 Creación de identidades de firma

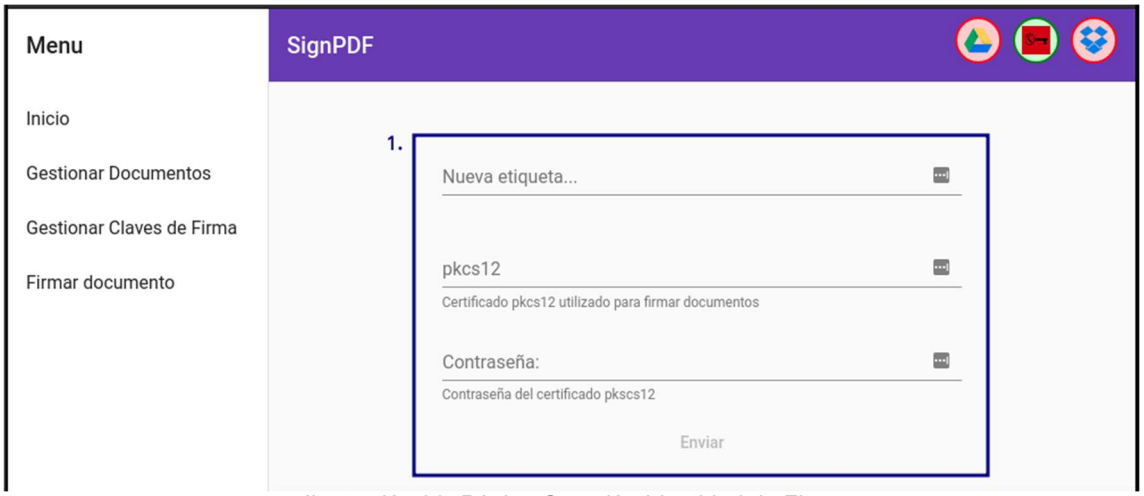


Ilustración 20: Página Creación Identidad de Firma

Esta vista contiene un solo componente que se compone de un formulario con los campos necesarios para crear una nueva identidad de firma.

4.2.7 Firma de documento

Esta vista es la más compleja de todas ya que contiene un formulario extenso en la que cada campo debe cumplir varias validaciones y además para que la firma se haga correctamente hace falta que el usuario haya autorizado a 'SignPDF' a varios servicios.

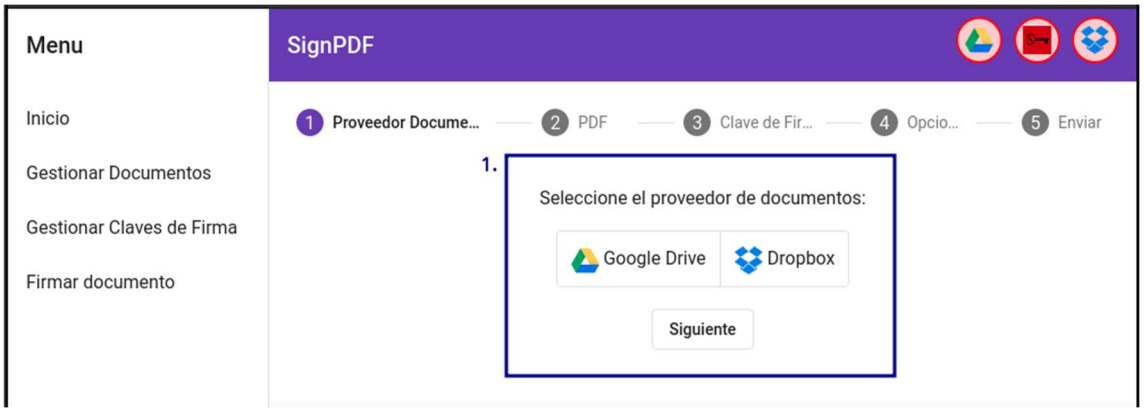


Ilustración 21: Página Firma 1

Este primer componente permite elegir el proveedor de documentos donde está alojado el documento PDF a firmar.

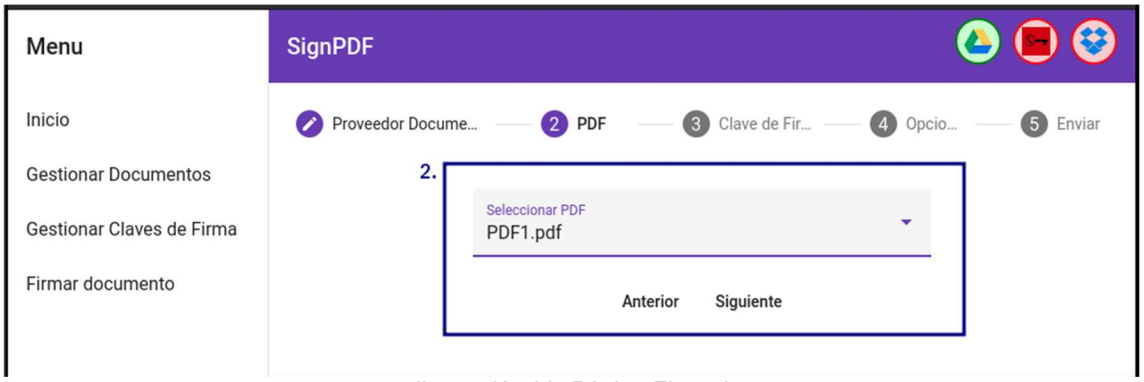


Ilustración 22: Página Firma 2

En este segundo componente se listan los documentos y se le pide al usuario que elija el PDF a firmar.

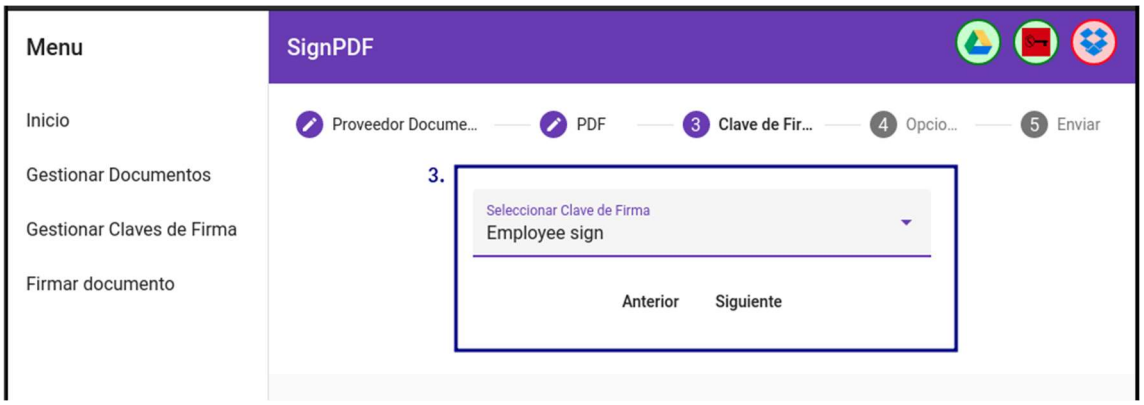


Ilustración 23: Página Firma 3

Del mismo modo que el componente anterior, se le pide al usuario que elija con qué identidad de firma desea firmar el PDF.

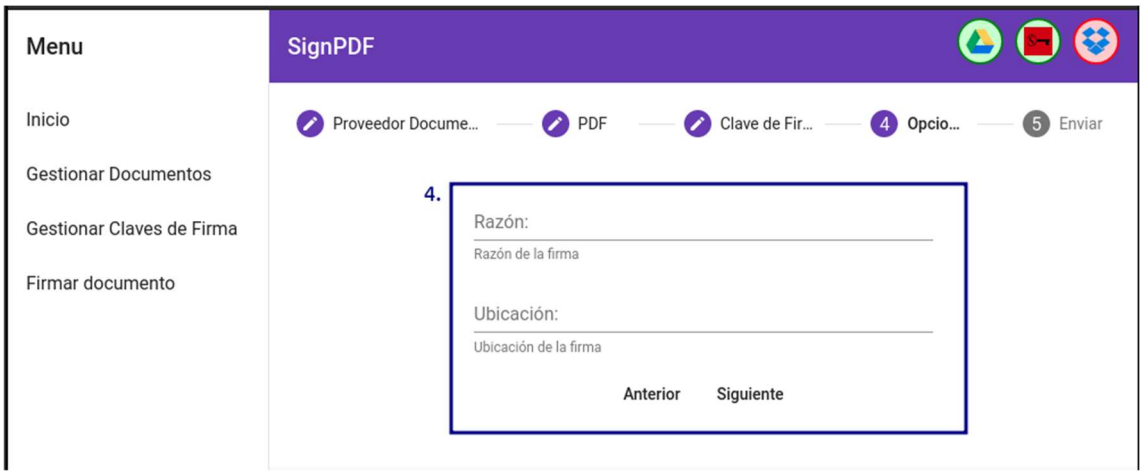


Ilustración 24: Página Firma 4

Para completar la firma es necesario que el usuario introduzca la razón de la firma y la ubicación de la misma, que serán más tarde incluidas en la firma a modo informativo.

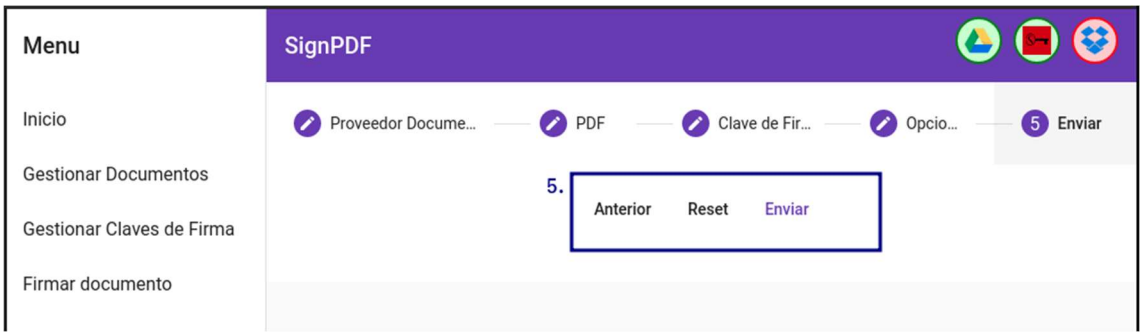


Ilustración 25: Página Firma 5

Por último, se incluye una vista de confirmación donde el usuario puede revisar los campos cumplimentados y enviarlo a la API si está todo de forma correcta.

Este formulario devuelve el PDF seleccionado firmado con la identidad de firma escogida, la firma se muestra en la esquina superior derecha como se puede ver en la imagen siguiente:

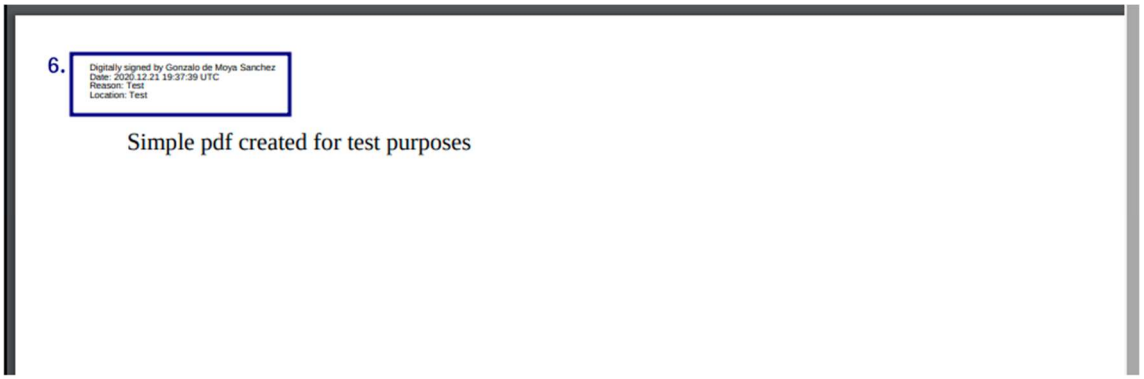


Ilustración 26: Ejemplo Documento Firmado

4.3 Infraestructura y despliegue

Aprovechando la fácil contenerización de las aplicaciones, se ha escogido desplegarlo en un entorno cloud basado en Kubernetes.

Cada una de las aplicaciones se ha construido en un contenedor Docker y se ha desplegado en un clúster gratuito de Kubernetes alojado en IBM Cloud con un solo nodo de 2 CPU y 4 GB de memoria.

La aplicación se ha expuesto a internet mediante un proxy nginx que enruta a los distintos microservicios o a los ficheros del *frontend* que están alojados de forma estática en el propio nginx.

Este nginx se ha expuesto mediante un servicio de Kubernetes de tipo Nodeport, por lo que la IP en la que está expuesta la aplicación es la del único nodo del clúster -159.122.175.6- y el puerto de Nodeport -31000-.

Para poder acceder de forma más cómoda a la aplicación se ha comprado un dominio gratuito - signpdf.ml - y se han configurado los DNS para que apunten a la IP donde está expuesta la aplicación.

Se ha creado un certificado SSL auto firmado para poder cifrar la comunicación mediante HTTPS, complemente necesario cuando se utilizan cookies de sesión por razones de seguridad.

A continuación, se detalla un diagrama de despliegue:

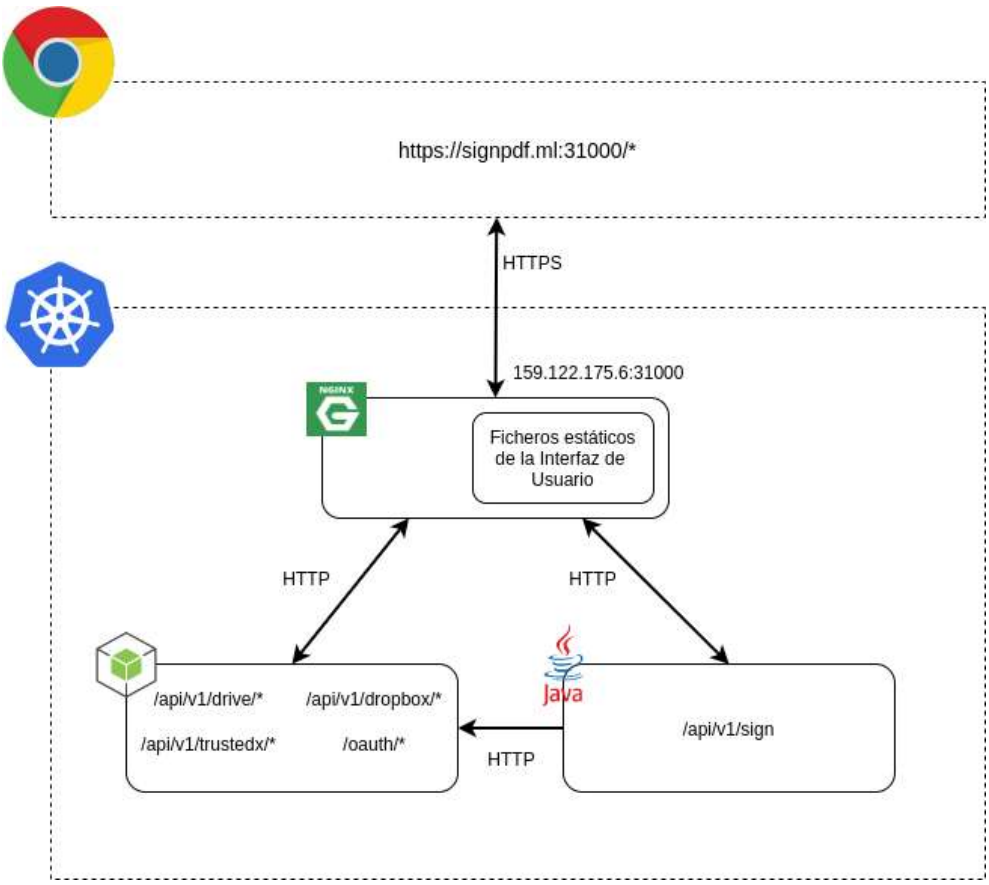


Ilustración 27: Diagrama Global Bajo Nivel

Como se puede comprobar en el esquema anterior, la única comunicación que se hace de forma cifrada es la del servidor con el proxy, una vez que la petición dentro del clúster de Kubernetes, se ha considerado que es un entorno seguro, y la comunicación se hace por HTTP.

5. Conclusiones

En el transcurso de este trabajo hemos comprobado las posibilidades que trae consigo la nueva regulación europea eIDAS con lo que respecta a la firma digital cualificada, facilitando mucho su uso y abriendo su utilización a la gran mayoría de la población con su próxima integración en el DNI 4.0.

Aunque se ha facilitado mucho su uso, la seguridad de este tipo de firmas no se ha visto afectada negativamente, e incluso, ha mejorado en ciertos aspectos como en la regulación de las máquinas confiables de firma.

En cuanto a los objetivos planteados, se han logrado todos los objetivos y todos los casos de uso han sido implementados tanto desde la API como desde la interfaz web e incluso se ha llegado más allá publicando la aplicación en un entorno cloud y haciéndola accesible desde Internet.

Respecto a la planificación, no se ha seguido de forma estricta, ya que en ciertos puntos se ha tenido que trabajar en varias tareas de forma simultánea debido a que no eran totalmente independientes, pero sí que ha servido como una guía a la hora de elegir las tareas más prioritarias a implementar y para evaluar si el desarrollo estaba siendo adecuado y no existían retrasos que pusieran en peligro la entrega del trabajo.

Referente a las líneas de trabajo futuro, lo podemos diferenciar en dos líneas principales, la primera sería el aumento de funcionalidades y aumento de proveedores de documentos y la otra sería trabajar para conseguir cumplir totalmente el reglamento europeo para que las firmas creadas en la plataforma sean completamente válidas.

En conclusión, se ha conseguido demostrar las ventajas que se consiguen delegando la custodia de las claves de firma a un servidor externo, donde está certificada la seguridad y privacidad de las claves y se aumenta su usabilidad, haciendo sencillo un proceso, que antes de la entrada en vigor de eIDAS, conllevaría mucho más esfuerzo.

Otra conclusión que hemos extraído de este trabajo es la importancia del estándar a la hora de implementar un servicio, en concreto nosotros hemos utilizado el estándar OAuth2 para la autorización de acceso y utilización de recursos externos por parte de la aplicación ‘SignPDF’. Gracias a esto hemos podido integrar toda la autorización en un simple fichero de configuración que contiene todos los datos necesarios para ejecutar la autorización, y evitarnos tener que implementar una autorización personalizada para cada uno de los servicios externos implementados.

4. Glosario

Agile: La metodología *agile* es un tipo de proceso de gestión de proyectos, utilizado principalmente para el desarrollo de software, donde las demandas y soluciones evolucionan conjuntamente.

Diagrama de Gantt: es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado [12].

HSM: Es un dispositivo informático físico que protege y administra claves criptográficas y proporciona procesamiento criptográfico. Un HSM es o contiene un módulo criptográfico [13].

OTP: son las siglas de contraseña de un solo uso, “One Time Password” en inglés, es una contraseña válida solo para una autenticación. OTP soluciona muchos problemas de seguridad que acompañan a métodos de autenticación simples.

5. Bibliografía

[1] España. Ley Orgánica 59/2003, de 19 de diciembre, de firma electrónica. Boletín Oficial del Estado. 20 de diciembre de 20031, núm. 304.

[2] Nota de prensa. Policía Nacional. Madrid, 04/11/2020. Disponible en: http://www.interior.gob.es/es/web/interior/noticias/detalle/-/journal_content/56_INSTANCE_1YSSl3xiWuPH/10180/12525577

[3] Clickup. Software de gestión de proyectos. Disponible en: https://clickup.com

[4] Unión Europea. Reglamento 910/2014, de 23 de julio de 2014, relativo a la identificación electrónica y los servicios de confianza para las transacciones electrónicas en el mercado interior.

[5] ETSI TS 102 778-1. Electronic Signatures and Infrastructures. PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES. European Telecommunications Standards Institute ETSI. 2009-07

[6] D. Hardt, Ed. Microsoft. RFC 6749. The OAuth 2.0 Authorization Framework. Octubre de 2012.

[7] TechCrunch. Google Drive will hit a billion users this week. Frederic Lardinois. Disponible en: https://techcrunch.com/2018/07/25/google-drive-will-hit-a-billion-users-this-week

[8] Industry Report. Dropbox Statistics, Users, Growth And Facts For 2020. Disponible en: https://saasscout.com/statistics/dropbox-statistics

[9] TrustedX. Plataforma de identificación, autenticación y firma electrónica centrada en el usuario para entornos Web. Disponible en: https://www.safelayer.com/es/productos/trustedx-electronic-signature

[10] Docusign. Descripción del producto. Disponible en: https://www.docusign.com.es/productos/firma-electronica

[11] International Organization for Standardization. (2008-07). Document management — Portable document format. Disponible en: https://www.iso.org/standard/51502.html

[12] Wikipedia. Diagrama_de_Gantt. Disponible en: https://es.wikipedia.org/wiki/Diagrama_de_Gantt

[13] NIST Special Publication 800-57 Part 2. Rev.1. Disponible en: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt2r1.pdf